ISO/IEC 29341-10-11

Edition 1.0 2008-11

INTERNATIONAL STANDARD

Information technology – UPnP Device Architecture – Part 10-11: Quality of Service Device Control Protocol – Quality of Service Manager Service

ISO/IEC 29341-10-11:2008(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: www.iec.ch

ISO/IEC 29341-10-11

Edition 1.0 2008-11

INTERNATIONAL STANDARD

Information technology – UPnP Device Architecture – Part 10-11: Quality of Service Device Control Protocol – Quality of Service Manager Service

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE Q

ICS 35.200

ISBN 2-8318-1011-71

CONTENTS

LIST OF TABLES

INFORMATION TECHNOLOGY –
UPNP DEVICE ARCHITECTURE –

Part 10-11: Quality of Service Device Control Protocol Quality of Service Manager Service

FOREWORD

1) ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards. Their preparation is entrusted to technical committees; any ISO and IEC member body interested in the subject dealt with may participate in this preparatory work. International governmental and non-governmental organizations liaising with ISO and IEC also participate in this preparation.

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

3) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC and ISO member bodies.

4) IEC, ISO and ISO/IEC publications have the form of recommendations for international use and are accepted by IEC and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

5) In order to promote international uniformity, IEC and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

6) ISO and IEC provide no marking procedure to indicate their approval and cannot be rendered responsible for any equipment declared to be in conformity with an ISO/IEC publication.

7) All users should ensure that they have the latest edition of this publication.

8) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC publication or any other IEC, ISO or ISO/IEC publications.

9) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

IEC and ISO draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of the putative patent rights. The holders of the putative patent rights have assured IEC and ISO that they are willing to negotiate free licences or licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of the putative patent rights are registered with IEC and ISO.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC and ISO shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29341-10-11 was prepared by UPnP Implementers Corporation and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of the ISO/IEC 29341 series, under the general title *Universal plug and play (UPnP) architecture*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

ORIGINAL UPNP DOCUMENTS (informative)

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

UPnP Document Title	ISO/IEC 29341 Part
UPnP Device Architecture 1.0	ISO/IEC 29341-1
UPnP Basic:1 Device	ISO/IEC 29341-2
UPnP AV Architecture:1	ISO/IEC 29341-3-1
UPnP MediaRenderer:1 Device	ISO/IEC 29341-3-2
UPnP MediaServer:1 Device	ISO/IEC 29341-3-3
UPnP AVTransport:1 Service	ISO/IEC 29341-3-10
UPnP ConnectionManager:1 Service	ISO/IEC 29341-3-11
UPnP ContentDirectory:1 Service	ISO/IEC 29341-3-12
UPnP RenderingControl:1 Service	ISO/IEC 29341-3-13
UPnP MediaRenderer:2 Device	ISO/IEC 29341-4-2
UPnP MediaServer:2 Device	ISO/IEC 29341-4-3
UPnP AV Datastructure Template:1	ISO/IEC 29341-4-4
UPnP AVTransport:2 Service	ISO/IEC 29341-4-10
UPnP ConnectionManager:2 Service	ISO/IEC 29341-4-11
UPnP ContentDirectory:2 Service	ISO/IEC 29341-4-12
UPnP RenderingControl:2 Service	ISO/IEC 29341-4-13
UPnP ScheduledRecording:1	ISO/IEC 29341-4-14
UPnP DigitalSecurityCamera:1 Device	ISO/IEC 29341-5-1
UPnP DigitalSecurityCameraMotionImage:1 Service	ISO/IEC 29341-5-10
UPnP DigitalSecurityCameraSettings:1 Service	ISO/IEC 29341-5-11
UPnP DigitalSecurityCameraStillImage:1 Service	ISO/IEC 29341-5-12
UPnP HVAC_System:1 Device	ISO/IEC 29341-6-1
UPnP HVAC_ZoneThermostat:1 Device	ISO/IEC 29341-6-2
UPnP ControlValve:1 Service	ISO/IEC 29341-6-10
UPnP HVAC_FanOperatingMode:1 Service	ISO/IEC 29341-6-11
UPnP FanSpeed:1 Service	ISO/IEC 29341-6-12
UPnP HouseStatus:1 Service	ISO/IEC 29341-6-13
UPnP HVAC_SetpointSchedule:1 Service	ISO/IEC 29341-6-14
UPnP TemperatureSensor:1 Service	ISO/IEC 29341-6-15
UPnP TemperatureSetpoint:1 Service	ISO/IEC 29341-6-16
UPnP HVAC_UserOperatingMode:1 Service	ISO/IEC 29341-6-17
UPnP BinaryLight:1 Device	ISO/IEC 29341-7-1
UPnP DimmableLight:1 Device	ISO/IEC 29341-7-2
UPnP Dimming:1 Service	ISO/IEC 29341-7-10
UPnP SwitchPower:1 Service	ISO/IEC 29341-7-11
UPnP InternetGatewayDevice:1 Device	ISO/IEC 29341-8-1
UPnP LANDevice:1 Device	ISO/IEC 29341-8-2
UPnP WANDevice:1 Device	ISO/IEC 29341-8-3
UPnP WANConnectionDevice:1 Device	ISO/IEC 29341-8-4
UPnP WLANAccessPointDevice:1 Device	ISO/IEC 29341-8-5
UPnP LANHostConfigManagement:1 Service	ISO/IEC 29341-8-10
UPnP Layer3Forwarding:1 Service	ISO/IEC 29341-8-11
UPnP LinkAuthentication:1 Service	ISO/IEC 29341-8-12
UPnP RadiusClient:1 Service	ISO/IEC 29341-8-13
UPnP WANCableLinkConfig:1 Service	ISO/IEC 29341-8-14
UPnP WANCommonInterfaceConfig:1 Service	ISO/IEC 29341-8-15
UPnP WANDSLLinkConfig:1 Service	ISO/IEC 29341-8-16
UPnP WANEthernetLinkConfig:1 Service	ISO/IEC 29341-8-17
UPnP WANIPConnection:1 Service	ISO/IEC 29341-8-18
UPnP WANPOTSLinkConfig:1 Service	ISO/IEC 29341-8-19
UPnP WANPPPConnection:1 Service	ISO/IEC 29341-8-20
UPnP WLANConfiguration:1 Service	ISO/IEC 29341-8-21
UPnP Printer:1 Device	ISO/IEC 29341-9-1
UPnP Scanner:1.0 Device	ISO/IEC 29341-9-2
UPnP ExternalActivity:1 Service	ISO/IEC 29341-9-10
UPnP Feeder:1.0 Service	ISO/IEC 29341-9-11
UPnP PrintBasic:1 Service	ISO/IEC 29341-9-12
UPnP Scan:1 Service	ISO/IEC 29341-9-13
UPnP QoS Architecture:1.0	ISO/IEC 29341-10-1
UPnP QosDevice:1 Service	ISO/IEC 29341-10-10
UPnP QosManager:1 Service	ISO/IEC 29341-10-11
UPnP QosPolicyHolder:1 Service	ISO/IEC 29341-10-12
UPnP QoS Architecture:2	ISO/IEC 29341-11-1
UPnP QOS v2 Schema Files	ISO/IEC 29341-11-2

UPnP Document Title	ISO/IEC 29341 Part
UPnP QosDevice:2 Service	ISO/IEC 29341-11-10
UPnP QosManager:2 Service	ISO/IEC 29341-11-11
UPnP QosPolicyHolder:2 Service	ISO/IEC 29341-11-12
UPnP RemoteUIClientDevice:1 Device	ISO/IEC 29341-12-1
UPnP RemoteUIServerDevice:1 Device	ISO/IEC 29341-12-2
UPnP RemoteUIClient:1 Service	ISO/IEC 29341-12-10
UPnP RemoteUIServer:1 Service	ISO/IEC 29341-12-11
UPnP DeviceSecurity:1 Service	ISO/IEC 29341-13-10
UPnP SecurityConsole:1 Service	ISO/IEC 29341-13-11

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0.

This service-type enables modeling of 'Quality of Service Manager' function capabilities. QosManager function is a combination of QosManager service and a control point functionality that discovers and controls QosDevice and QosPolicyHolder services running on the network. QosManager function is responsible for requesting, updating, releasing and in general controlling the Quality of Service assigned by networking devices to various traffic streams. The QosManager service will be invoked from a UPnP Control Point to perform the functions related to setting up Quality of Service for that traffic. Once the network is configured with respect to the Quality of Service for the upcoming traffic stream, the QosManager service will hand back control to the control point. This service provides a mechanism for Control Points to:

- Be agnostic of the Quality of Service capabilities and associated details about the various devices on the network.
- Hand-over the tasks of setting up, modifying and revoking the Quality of Service associated with every traffic stream
- Hand-over the task of control point functionality with respect to the UPnP QosDevice service – its discovery, control and eventing.
- Hand-over the task of control point functionality with respect to the UPnP QosPolicyHolder service – its discovery, control and eventing.

Thus, a QosManager is a dual-role entity in the sense that it exposes QosManager service to the Control Point while acting as a Control Point for the QosPolicyHolder and QosDevice services running on the network. This document describes the components of the UPnP QosManager service.

- This service template does not address:

  The control point functionality within a QosManager. However, useful information about the control point functionality in the QosManager can be found in the following documents:

  - UPnP QoS Architecture document
  - UPnP QosDevice Service Definition Document
  - UPnP QosPolicyHolder Service Definition Document

1.1. Referenced Specifications

Unless explicitly stated otherwise herein, implementation of the mandatory provisions of any standard referenced by this specification shall be mandatory for compliance with this specification.

This section lists the normative references used in this document and includes the tag inside square brackets that is used for each sub reference:

[IEEE] - IEEE 802.1D-2004, Annex G, *IEEE Standard for Information technology – Telecommunications and information exchange between systems - IEEE standard for local and metropolitan area networks, Common specifications - Media access control (MAC) Bridges*, 2004.

[XML] – *Extensible Markup Language (XML) 1.0 (Second Edition)*, T. Bray, J.Paoli, C. M. Sperberg McQueen, E Maler, eds. W3C Recommendations, 6 October 2000.

[DEVICE] - *UPnP Device Architecture, version 1.0.*

[POLICY HOLDER] – UPnP QosPolicyHolder:1 Service Document.

[QOS DEVICE] – UPnP QosDevice:1 Service Document.

2. Service Modeling Definitions

2.1. ServiceType

The following service type identifies a service that is compliant with this template:

xmlns="urn:schemas-upnp-org:service:QosManager:1"

xmlns=http://www.upnp.org/schemas/TrafficDescriptorv1.xsd

2.2. State Variables

Reader Note: For first-time reader, it may be more insightful to read the action definitions before reading the state variable definitions.

2.2.1.1. XML Fragments as UPnP Arguments

The UPnP QoS Framework often uses XML Fragments as arguments in UPnP actions. The containing UPnP data type is a string. This places restrictions on a string's content; it has to represent a well-formed XML fragment (this includes a complete XML document).

The XML schemas used in UPnP-QoS are defined in the respective files located on `http://www.upnp.org/schemas/`

In their XML fragments, implementations may use an explicit reference to appropriate name spaces.

At several places in the XML schemas there is room for vendor differentiation through the use of the "any"-tag. When extending UPnP-QoS with their own XML tags, vendors should use a name space to prevent collisions of their tags with those of other vendors. It is recommended that implementations are not required to retrieve the corresponding schemas from the Internet.

Finally, an XML fragment, in adherence to the UPnP Device Architecture 1.0 [DEVICE] , needs to be escaped by using the normal XML rules, [XML] Section 2.4 Character Data and Markup, before embedding it in a SOAP request or response message. The XML escaping rules are summarized from the [XML] reference mentioned above:

- The (<) character is encoded as (&lt;)
- The (>) character is encoded as (&gt;)
- The (&) character is encoded as (&amp;)
- The (") character is encoded as (&quot;)
- The (') character is encoded as (&apos;)

Table 2-1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
A_ARG_TYPE_TrafficDescriptor	R	string	Escaped XML fragment of trafficDescriptor Schema (See Appendix A for details).	n/a	n/a
A_ARG_TYPE_TrafficHandle	R	string		n/a	n/a
A_ARG_TYPE_NumTrafficDescriptors	R	integer	Non-negative integer.	n/a	n/a
A_ARG_TYPE_NumPolicyHolders	R	integer	Non-negative integer.	n/a	n/a
A_ARG_TYPE_ ListOfTrafficDescriptors	R	string	Escaped XML fragment	n/a	n/a

[1] R = Required, O = Optional, X = Non-standard.

[2] Values listed in this column are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

2.2.2. A_ARG_TYPE_TrafficDescriptor

This state variable contains information about a particular traffic stream. A Traffic Descriptor consists of an XML structure consisting of:

- TrafficHandle
- TrafficID
- AvailableOrderedTspecList
- QosBoundarySourceAddress
- TrafficImportanceNumber
- QosBoundaryDestinationAddress
- UserName
- CpName
- VendorApplicationName
- PortName
- ServiceProviderServiceName
- TrafficLeaseTime

Please refer to the detailed TrafficDescriptor XML schema in Appendix A. Here are some details about information in Traffic Descriptor:

Tspec

Tspec contains a description of Content QoS Requirements. Tspec is represented in the form of a XML structure. In UPnP AV scenario, this information is extracted from the Content Directory Service of the Media Server. In the Content Directory Service, Tspec is represented either as a string containing escaped XML structure, or an URI pointing to the escaped XML structure. UPnP AV Control Point uses CDS:Browse and/or CDS:Search action calls to acquire the Tspec(s) associated with the content and creates an AvailableOrderedTspecList field in the Traffic Descriptor.

Tspec XML structure consists of the following:

- ❑ TspecIndex: Unique numerical index associated with a particular Tspec. The value of Tspecindex indicates preference (as defined by the application or the end user). A Tspec with smaller Index is more preferred compared to a Tspec with larger Index.
- ❑ AVTransportURI: This string field contains a unique URI associated with the UPnP AV content item for which QoS is being requested.

- ❑ AVTransportInstanceID: This integer field contains a unique Instance ID associated with the UPnP AV Transport service associated with the content item for which QoS is being requested.
- ❑ TrafficClass: The traffic class associated with the traffic stream. This is an enumerated variable that can be assigned to one of the following list of values:
  - o Network Control
  - o Streaming Control
  - o Voice
  - o AV
  - o Data
  - o Audio
  - o Image
  - o Gaming
  - o Other
  - o Background

TspecIndex must be filled in by the Control Point. Other Tspec parameters may be omitted if they are not known. TspecIndexs need not be consecutive numbers. The Tspec with lower TspecIndex is more preferred than a higher TspecIndex. In this example, the Tspec with 2 is more preferred than that with 300. The order of Tspec structures in the AvailableOrderedTspecList is not relevant. All Tspecs must have unique TspecIndex value within an AvailableOrderedTspecList.

```
<AvailableOrderedTspecList>
     <Tspec>
          <TspecIndex>300</TspecIndex>
          <TrafficClass>AV</TrafficClass>
     </Tspec>
     <Tspec>
          <TspecIndex>2</TspecIndex>
          <TrafficClass>Audio</TrafficClass>
     </Tspec>
</AvailableOrderedTspecList>
```

TrafficHandle

TrafficHandle is a unique identifier associated with a particular instance of TrafficDescriptor; i.e. a particular traffic stream. It is a unique string generated by the QosManager and provided to the Control Point in response to the RequestTrafficQos action. The QosManager must insure that the generated Traffic Handle string is unique for all Traffic Descriptors on the network. In all subsequent communication between the CP and the QosManager service, TrafficHandle is used to reference a particular traffic stream.

AvailableOrderedTspecList

Contains one or more Tspec components. The Tspec index reflects the order of preference. Tspecs with smaller index values are more preferred. AvailableOrderedTspecList is required. A Tspec may be defined via any combination of its components.

ActiveTspecIndex

Active TspecIndex contains an integer which indicates the index of the current active Tspec from the TspecList. Tspec and Tspec List are defined above.

QosBoundarySourceAddress

If a traffic stream originates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams originating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual source address. QosBoundarySourceAddress is used by QosManager for decisions related to path determination and device selection.

QosBoundaryDestinationAddress

If a traffic stream terminates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams terminating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual destination address. QosBoundaryDestinationAddress is used by QosManager for decisions related to path determination and device selection.

TrafficId

TrafficId contains information about identifying / classifying packets that belong to that particular traffic stream. It consists of an XML structure consisting of: SourceIP; DestinationIP; SourcePort; DestinationPort; Protocol.

Any of these arguments that are not known are represented as null.

UserName

UserName field contains a string associated with the user who requested a particular traffic stream. This field, if present, shall identify the name of the user initiating the UPnP QoS Action. The field must be <= 64 UTF-8 characters. At present this specification does not define the semantics for this field. This field represents the name of the user initiating the QoS action. Examples include, “Dad”, “Jimmy”, “CN#1234567”. The latter is an example of a name provided by a service provider to uniquely identify its individual customers.

CpName

CpName field contains a string associated with the Control Point requesting the traffic stream. This field, if present, shall identify the name of the control point initiating the UPnP QoS Action. The field must be <= 64 UTF-8 characters. At present this specification does not define the semantics for this field. The CpName may specify the brand name of the control point or it may indicate the location of the control point such as “living room”

VendorApplicationName

VendorApplicationName field contains a single URI string associated with an application. This field, if present, shall identify the name of the application initiating the UPnP QoS Action. For applications specified by vendors, the value of this field must begin with “urn:”,followed by an ICANN domain name owned by the vendor, followed by “:application:”, followed by an application name. i.e. the value of this field should be constructed as follows:

urn:*domain-name*:application:*appl-name*

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

appl-name: Name of the application provided by a vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *appl-name* field must be <= 64 UTF-8 characters. The *application-name* must not contain the colon character.

When requesting QoS it may be useful to know the manufacturer and name of the application software client or server that will process the traffic stream.

PortName:

PortName field contains a single URI string associated with the fixed port used by an application for its connection set-up. This field, if present, shall identify the port number used by the application. The value of this field must begin with “urn:”, followed by an ICANN domain name owned by the vendor, followed by “:port:”, followed by the port name. i.e. the value of this field should be constructed as follows

urn:*domain-name*:port:*port-number*

domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

port-number: A fixed port number used by the application either for connection set-up. This must be an integer value in the range 1 to 65535. If the application uses a range of port numbers then the starting port number of that range shall be specified. This value is either a vendor specific port used for the application or an IANA assigned port number for the application. The list of all the IANA assigned port numbers is maintained by IANA and is available at http://www.iana.org/assignments/port-numbers.

These fields indicate the port that is used by an application for its connection set-up. The use of IANA assigned port numbers is encouraged, but if not available for a particular application, a vendor assigned port number may be used.

ServiceProviderServiceName

ServiceProviderServiceName field contains a single URI string associated with a service offered by a service provider. This field, if present, shall identify the name of the service offered by a service provider.. For services provided by service providers, the value of this field must begin with “urn:”, followed by an ICANN domain name owned by a service provider , followed by “service:”, followed by a service name. i.e. the value of this field should be constructed as follows:

urn:*domain-name*:service:*serv-name*

domain-name: Service Provider domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

serv-name: Name of the service provided by a service vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The serv-*name* field must be <= 64 UTF-8 characters. The serv-*name* must not contain the colon character.

This represents the name of a particular service being offered by a service provider. Note that while multiple services provided by the same service provider may use the same software application, different Service offerings may be given different priorities (traffic importance).

TrafficLeaseTime

This field contains the lease-time associated with a particular traffic stream. The lease time is expressed in ‘Seconds’. When not specified, it indicates an indefinite lease.

MediaServerConnectionId:

This is an optional field. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaServerConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details. The connection Id is obtained from the Media Server via CM::PrepareForConnection action.

MediaRendererConnectionId:

This is an optional field. This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaRendererConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details. The connection Id is obtained from the Media Renderer via CM::PrepareForConnection action.

Use the following XML documents as examples for TrafficDescriptor schema.

Example 1:

```
<TrafficDescriptor
  xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd">
  <TrafficHandle>kiwin1</TrafficHandle>
  <TrafficId>
    <SourceAddress>
      <Ipv4>192.168.1.50</Ipv4>
    </SourceAddress>
    <SourcePort>23</SourcePort>
    <DestinationAddress>
      <Ipv4>192.168.1.53</Ipv4>
    </DestinationAddress>
    <DestinationPort>23</DestinationPort>
    <IpProtocol>1</IpProtocol>
  </TrafficId>
  <AvailableOrderedTspecList>
    <Tspec>
      <TspecIndex>300</TspecIndex>
      <TrafficClass>AV</TrafficClass>
    </Tspec>
    <Tspec>
      <TspecIndex>2</TspecIndex>
      <TrafficClass>Audio</TrafficClass>
    </Tspec>
  </AvailableOrderedTspecList>
  <ActiveTspecIndex>300</ActiveTspecIndex>
  <TrafficImportanceNumber>5</TrafficImportanceNumber>
  <OptionalPolicyParams>
    <CpName>Amy's CP</CpName>
  </OptionalPolicyParams>
</TrafficDescriptor>
```

Example 2:

```
<TrafficDescriptor
  xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:prv="http://myPrivate.com"
  xmlns:prv2="http://myPrivate2.com"
  xsi:schemaLocation="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd">

  <TrafficHandle>kiwin1</TrafficHandle>
  <TrafficId>
    <SourceAddress>
      <Ipv4>192.168.1.50</Ipv4>
    </SourceAddress>
    <SourcePort>23</SourcePort>
    <DestinationAddress>
      <Ipv4>192.168.1.53</Ipv4>
    </DestinationAddress>
    <DestinationPort>23</DestinationPort>
    <IpProtocol>1</IpProtocol>
  </TrafficId>
  <AvailableOrderedTspecList>
    <Tspec>
      <TspecIndex>300</TspecIndex>
      <TrafficClass>AV</TrafficClass>
    </Tspec>
    <Tspec>
      <TspecIndex>2</TspecIndex>
      <TrafficClass>Audio</TrafficClass>
    </Tspec>
  </AvailableOrderedTspecList>
  <ActiveTspecIndex>300</ActiveTspecIndex>
  <TrafficImportanceNumber>5</TrafficImportanceNumber>
  <OptionalPolicyParams>
    <CpName>Amy's CP</CpName>
  </OptionalPolicyParams>
</TrafficDescriptor>
```

2.2.3. A_ARG_TYPE_TrafficHandle

This state variable contains a unique identifier associated with a particular instance of TrafficDescriptor i.e. a particular traffic stream. It is a unique string generated by the QosManager and provided to the Control Point in response to the RequestTrafficQos action. In all subsequent communication between the CP and the QosManager service, TrafficHandle is used to reference a particular traffic stream. QosManager must insure that TrafficHandle is a unique string that identifies one and only one TrafficDescriptor on the network.

2.2.4. A_ARG_TYPE_NumTrafficDescriptors

This state variable contains the number of Traffic Descriptors registered / admitted by a given instance of the QosManager service. This information is returned in response to the BrowseAllTrafficDescriptors service action call from the Control Point.

2.2.5. A_ARG_TYPE_NumPolicyHolders

This state variable contains the number of instances of QosPolicyHolder service discovered by a given instance of the QosManager service. This variable is returned in response to RequestTrafficQos and UpdateTrafficQos actions (as an output argument) to convey to a Control Point the number of active Policy Holders on the network. If the value of this variable is not equal to 1, it means that the default policy was used by the QosManager to make admission control decisions.

2.2.6. A_ARG_TYPE_ ListOfTrafficDescriptors

This state variable contains one or more instances of the state variable A_ARG_TYPE_TrafficDescriptor. Here is the XML schema for this state variable:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
targetNamespace="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  xmlns="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:td="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
  elementFormDefault="qualified" id="ListOfTrafficDescriptors">

  <xs:import
namespace="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
   schemaLocation="TrafficDescriptorv1.xsd"/>
  <xs:annotation>
    <xs:documentation xml:lang="en">
      List of Traffic Descriptors schema.
      Copyright 2004, 2005 UPnP(tm). All rights reserved.
    </xs:documentation>
  </xs:annotation>
  <xs:element name="ListOfTrafficDescriptors"
type="ListOfTrafficDescriptorsType"/>
  <xs:complexType name="ListOfTrafficDescriptorsType">
    <xs:sequence>
      <xs:element name="TrafficDescriptor" type="td:TrafficDescriptorType"
minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Use the following XML document as an example for ListOfTrafficDescriptors schema.

```
<ListOfTrafficDescriptors
  xmlns="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.upnp.org/schemas/ListOfTrafficDescriptors.xs
d">
  <TrafficDescriptor>
    <TrafficHandle>kiwin1</TrafficHandle>
    <TrafficId>
```

```
      <SourceAddress>
        <Ipv4>192.168.1.50</Ipv4>
      </SourceAddress>
      <SourcePort>23</SourcePort>
      <DestinationAddress>
        <Ipv4>192.168.1.50</Ipv4>
      </DestinationAddress>
      <DestinationPort>23</DestinationPort>
      <IpProtocol>1</IpProtocol>
    </TrafficId>
    <AvailableOrderedTspecList>
      <Tspec>
        <TspecIndex>300</TspecIndex>
        <TrafficClass>AV</TrafficClass>
      </Tspec>
      <Tspec>
        <TspecIndex>2</TspecIndex>
        <TrafficClass>Audio</TrafficClass>
      </Tspec>
    </AvailableOrderedTspecList>
    <ActiveTspecIndex>1</ActiveTspecIndex>
    <TrafficImportanceNumber>5</TrafficImportanceNumber>
    <OptionalPolicyParams>
      <CpName>Amy's CP</CpName>
    </OptionalPolicyParams>
  </TrafficDescriptor>
  <TrafficDescriptor>
    <TrafficHandle>kiwin2</TrafficHandle>
    <TrafficId>
      <SourceAddress>
        <Ipv4>192.168.1.53</Ipv4>
      </SourceAddress>
      <SourcePort>23</SourcePort>
      <DestinationAddress>
        <Ipv4>192.168.1.55</Ipv4>
      </DestinationAddress>
      <DestinationPort>23</DestinationPort>
      <IpProtocol>1</IpProtocol>
    </TrafficId>
    <AvailableOrderedTspecList>
      <Tspec>
        <TspecIndex>300</TspecIndex>
        <TrafficClass>AV</TrafficClass>
      </Tspec>
      <Tspec>
        <TspecIndex>2</TspecIndex>
        <TrafficClass>Audio</TrafficClass>
      </Tspec>
    </AvailableOrderedTspecList>
    <ActiveTspecIndex>1</ActiveTspecIndex>
    <TrafficImportanceNumber>5</TrafficImportanceNumber>
    <OptionalPolicyParams>
      <CpName>Amy's CP</CpName>
    </OptionalPolicyParams>
  </TrafficDescriptor>
</ListOfTrafficDescriptors>
```

2.2.7. Relationships Between State Variables

2.3. Eventing and Moderation

Table 2-2: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
A_ARG_TYPE_TrafficDescriptor	*NO*	*n/a*	*n/a*	*n/a*	*n/a*
A_ARG_TYPE_TrafficHandle	*NO*	*n/a*	*n/a*	*n/a*	*n/a*
A_ARG_TYPE_NumTrafficDescriptors	*NO*	*n/a*	*n/a*	*n/a*	*n/a*
A_ARG_TYPE_NumPolicyHolders	*NO*	*n/a*	*n/a*	*n/a*	*n/a*
A_ARG_TYPE_ ListOfTrafficDescriptors	*NO*	*n/a*	*n/a*	*n/a*	*n/a*

[1] Determined by N, where Rate = (Event)/(N secs).

[2] (N) * (allowedValueRange Step).

2.3.1. Event Model

2.4. Actions

Immediately following this table is detailed information about these actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 2-3: Actions

Name	Req. or Opt. [1]
RequestTrafficQos	R
UpdateTrafficQos	R
ReleaseTrafficQos	R
BrowseAllTrafficDescriptors	R

[1] R = Required, O = Optional, X = Non-standard.

2.4.1 RequestTrafficQos

Control Point can invoke this action for setting up the Quality of Service for a particular traffic stream. At least one of the elements of SourceAddress or DestinationAddress or SourcePort or DestinationPort or IpProtocol must be specified in the TrafficId structure.

If Control Point supplies TrafficImportanceNumber in TrafficDescriptor to QosManager when calling this action QosManager must return error 721.

If Control Point supplies TrafficDescriptor to QosManager when calling this action, the QosManager must return error code 722.

QosManager must include a valid ActiveTspecIndex value in the TrafficDescriptor when it returns a non error value in response to this action.

If Control Point supplies a non-null TrafficHandle to QosManager when calling this action, the QosManager must return error code 701.

When there are multiple Tspec(s) in the TrafficDescriptor provided by the Control Point to RequestTrafficQoS at the QosManager, the QosManager must set up the most preferred Tspec first. If the setup fails, then the QosManager must try next preferred Tspec and so on.

2.4.1.1. Arguments

Table 2-4: Arguments for RequestTrafficQos

Argument	Direction	relatedStateVariable
InitialTrafficDescriptor	In	A_ARG_TYPE_TrafficDescriptor
TrafficHandle	Out	A_ARG_TYPE_TrafficHandle
NumPolicyHolders	Out	A_ARG_TYPE_NumPolicyHolders
UpdatedTrafficDescriptor	Out	A_ARG_TYPE_TrafficDescriptor

2.4.1.2. Dependency on State (if any)

Since UPnP QosManager is defined as a stateless entity, there is no effect of this action on state. However, it is expected that since UPnP QosManager is a dual-role entity, it would access the interface exposed by UPnP QosDevice service hosted on various UPnP QoS enabled devices and attempt to request QoS for the specified traffic stream.

2.4.1.3. Effect on State (if any)

Since UPnP QosManager is defined as a stateless entity, there is no effect of this action on state. However, it is expected that since UPnP QosManager is a dual-role entity, it would access the interface exposed by UPnP QosDevice service hosted on various UPnP QoS enabled devices and attempt to request QoS for the specified traffic stream.

2.4.1.4. Errors

Table 2-5: Error Codes for RequestTrafficQos

errorCode	errorDescription	Description
701	TrafficHandle must not be specified by the Control Point	
711	TrafficId could not be completed	The QosManager could not complete the TrafficId
713	Malformed Name-string	The string VendorApplicationName, ServiceProviderServiceName, … does not follow the format from the specification
721	TrafficImportanceNumber must not be specified by the Control Point	
722	The ActiveTspecIndex must not be specified by the Control Point	
740	No QosDevice at Source IP address	Source QosDevice does not exist.
741	No QosDevice at Destination IP address	Destination QosDevice does not exist.
742	No QosDevice at QosBoundarySourceAddress	Source Boundary QosDevice does not exist.
743	No QosDevice at QosBoundaryDestinationAddress	Destination Boundary QosDevice does not exist.
744	No QosDevices available	No QosDevice instances found on the network.
746	QosPolicyHolder failure	QosManager gets incorrect information from the QosPolicyHolder (could be malformed XML)
752	QosBoundarySourceAddress or QosBoundaryDestinationAddress not on the QosManaged subnet	The source and/or destination is not on the subnet which is under the management of this QosManager
762	A QosDevice is not capable of handling this connection	The maximum bit rate exceeds the capacity of the interfaces of the QosDevice.
770	Network is not cooperating	A QosDevice service is not responding
771	Network does not allow temporary	Existing traffic streams have higher or equal UserImportanceNumber. Hence the traffic stream cannot be setup at this time
772	Network not capable (never)	Network is unable to ever setup QoS for this traffic stream. e.g. No path exists from source to sink.

errorCode	errorDescription	Description
773	Cannot retrieve XML namespace	An XML namespace was included in one of the arguments, but the QosManager service was unable to retrieve the namespace from the location provided.

2.4.2. UpdateTrafficQos

Control Point can invoke this action for updating the Quality of Service for a particular traffic stream.

If Control Point supplies TrafficImportanceNumber in TrafficDescriptor to QosManager when calling this action QosManager must return error 721.

If Control Point supplies TrafficDescriptor to QosManager when calling this action, the QosManager must return error code 722.

QosManager must include a valid ActiveTspecIndex value in the TrafficDescriptor when it returns a non error value in response to this action.

If the Control Point does not supply a TrafficHandle when calling this action the QosManager must return error code 700.

The Control Point may update TrafficLeaseTime, UserName and/or CpName during UpdateTrafficQos action on the QosManager. All other non-any parameters must not be updated by the Control Point, and such an input must result in error code 714.

In response to the Control Point calling the UpdateTrafficQos action, the QosManager may update ActiveTspecIndex, and/or TrafficImportanceNumber in the TrafficDescriptor structure. All other non-any parameters must not be updated by the QosManager and result in Error Code 714.

2.4.2.1. Arguments

Table 2-6: Arguments for UpdateTrafficQos

Argument	Direction	relatedStateVariable
TrafficHandle	In	A_ARG_TYPE_TrafficHandle
RequestedTrafficDescriptor	In	A_ARG_TYPE_TrafficDescriptor
ImplementedTrafficDescriptor	Out	A_ARG_TYPE_TrafficDescriptor
NumPolicyHolders	Out	A_ARG_TYPE_NumPolicyHolders

2.4.2.2. Dependency on State (if any)

Since UPnP QosManager is defined as a stateless entity, there is no dependency for this action on the state.

2.4.2.3. Effect on State (if any)

Since UPnP QosManager is defined as a stateless entity, there is no effect of this action on state. However, it is expected that since UPnP QosManager is a dual-role entity, it would access the interface exposed by UPnP QosDevice service hosted on various UPnP QoS enabled devices and attempt to update the specified traffic stream.

2.4.2.4. Errors

Table 2-7: Error Codes for UpdateTrafficQos

errorCode	errorDescription	Description
703	Traffic Handle unknown to this device	
713	Malformed Name-string	The string VendorApplicationName, ServiceProviderServiceName, … does not follow the format from the specification
714	Tried to update unmodifiable Traffic Descriptor elements	QosManager cannot be requested to update some of the TD parameters e.g. TrafficId, TrafficHandle
740	No QosDevice at Source IP address	Source QosDevice does not exist.
741	No QosDevice at Destination IP address	Destination QosDevice does not exist.
742	No QosDevice at QosBoundarySource Address	Source Boundary QosDevice does not exist.
743	No QosDevice at QosBoundaryDestinationAddress	Destination Boundary QosDevice does not exist.
744	No QosDevices available	No QosDevice instances found on the network.
746	QosPolicyHolder failure	QosManager gets incorrect information from the QosPolicyHolder (could be malformed XML)
752	QosBoundarySource Address or QosBoundaryDestinationAddress not on the QosManaged subnet	The source and/or destination is not on the subnet which is under the management of this QosManager
770	Network is not cooperating	A QosDevice service instance is not responding
771	Network does not allow temporary	Existing traffic streams have higher or equal UserImportanceNumber. Hence the traffic stream cannot be setup at this time
773	Cannot retrieve XML namespace	An XML namespace was included in one of the arguments, but the service was unable to retrieve the namespace from the location provided.

2.4.3. ReleaseTrafficQos

Control Point can invoke this action for releasing the Quality of Service for a particular traffic stream.

2.4.3.1. Arguments

Table 2-8: Arguments for ReleaseTrafficQos

Argument	Direction	relatedStateVariable
RevokeTrafficHandle	In	A_ARG_TYPE_TrafficHandle

2.4.3.2. Dependency on State (if any)

Since UPnP QosManager is defined as a stateless entity, there is no dependency for this action on the state.

2.4.3.3. Effect on State (if any)

Since UPnP QosManager is defined as a stateless entity, there is no effect of this action on state. However, it is expected that since UPnP QosManager is a dual-role entity, it would access the interface exposed by UPnP QosDevice service hosted on various UPnP QoS enabled devices and attempt to revoke the specified traffic stream.

2.4.3.4. Errors

Table 2-9: Error Codes for ReleaseTrafficQos

errorCode	errorDescription	Description
703	Traffic Handle unknown to this device	
744	No QosDevices available	
770	Network is not cooperating	A QosDevice service instance is not responding
773	Cannot retrieve XML namespace	An XML namespace was included in one of the arguments, but the service was unable to retrieve the namespace from the location provided.

2.4.4. BrowseAllTrafficDescriptors

Control Point can invoke this action to browse all the 'TrafficDescriptors' configured on the network. Each TrafficDescriptor represents Quality of Service for a particular traffic stream.

2.4.4.1. Arguments

Table 2-10: Arguments for BrowseAllTrafficDescriptors

Argument	Direction	relatedStateVariable
NumberOfTrafficDescriptors	Out	A_ARG_TYPE_NumTrafficDescriptors
TrafficDescriptorList	Out	A_ARG_TYPE_ ListOfTrafficDescriptors

2.4.4.2. Dependency on State (if any)

This action lets the Control Point learn about the currently defined UPnP QoS TrafficDescriptors. This action does not have any dependency on state.

2.4.4.3. Effect on State (if any)

This action lets the Control Point learn about the currently defined UPnP QoS TrafficDescriptors. This action does not have any effect on state.

2.4.4.4. Errors

Table 2-11: Error Codes for BrowseAllTrafficDescriptors

errorCode	errorDescription	Description
704	Network contains Different TrafficDescriptors for same TrafficHandle	

2.4.5. Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the section on Description).

2.4.6. Relationships Between Actions

2.4.7. Common Error Codes

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error must be returned. These common error codes have already been defined in UPnP Device Architecture and other Technical Committee documents.

Table 2-12: Common Error Codes

errorCode	errorDescription	Description
400-499		See UPnP Device Architecture section on Control.
500-599		See UPnP Device Architecture section on Control.
600-699		See UPnP Device Architecture section on Control.
701	TrafficHandle must not be specified by the Control Point	
703	Traffic Handle unknown to this device	
704	Network contains Different TrafficDescriptors for same TrafficHandle	
711	TrafficId could not be completed	The QosManager could not complete the TrafficId
713	Malformed Name-string	The string VendorApplicationName, ServiceProviderServiceName, … does not follow the format from the specification
714	Tried to update unmodifiable Traffic Descriptor elements	QosManager cannot be requested to update some of the TD parameters e.g. TrafficId, TrafficHandle
721	TrafficImportance Number must not be specified by the Control Point	
722	The ActiveTspecIndex must not be specified by the Control Point	
740	No QosDevice at Source IP address	Source QosDevice does not exist.
741	No QosDevice at Destination IP address	Destination QosDevice does not exist.
742	No QosDevice at QosBoundarySourceAddress	Source Boundary QosDevice does not exist.

errorCode	errorDescription	Description
743	No QosDevice at QosBoundaryDestinationAddress	Destination Boundary QosDevice does not exist.
744	No QosDevices available	No QosDevice instances found on the network.
746	QosPolicyHolder failure	QosManager gets incorrect information from the QosPolicyHolder (could be malformed XML)
752	QosBoundarySourceAddress or QosBoundaryDestinationAddress not on the QosManaged subnet	The source and/or destination is not on the subnet which is under the management of this QosManager
762	A QosDevice is not capable of handling this connection	The maximum bit rate exceeds the capacity of the interfaces of the QosDevice.
770	Network is not cooperating	A QosDevice service is not responding
771	Network does not allow temporary	Existing traffic streams have higher or equal UserImportanceNumber. Hence the traffic stream cannot be setup at this time
772	Network not capable (never)	Network is unable to ever setup QoS for this traffic stream. e.g. No path exists from source to sink.
773	Cannot retrieve XML namespace	An XML namespace was included in one of the arguments, but the service was unable to retrieve the namespace from the location provided.

2.5. Theory of Operation

This service-type enables modeling of 'Quality of Service Manager' capabilities. A UPnP QosManager is responsible for setting up, updating, revoking and in general controlling the Quality of Service assigned by networking devices to various traffic streams. A QosManager acts as a control point to UPnP QosDevice services. A QosManager discovers the QosDevice services on the local area network, and may use 'QD:GetQosDeviceCapabilities'[2] and 'QD:GetQostate' actions exposed by the QosDevice service to get information about the device. 'QD:GetQosDeviceCapabilities' action returns capabilities of QoS enabled devices (and the details on the various network interfaces present on the device). 'QD:GetQosState' action returns the current state of the UPnP QoS enabled device with information such as the TrafficDescriptors that are currently active.

The UPnP QosManager exposes the 'QM:BrowseAllTrafficDescriptors'[3] action to the control point. Upon receiving this action request, the QosManager can report back the details on various traffic streams admitted on the UPnP QoS devices on the network. QosManager also discovers the QosPolicyHolder services on the local area network. It is expected that there will be one such PolicyHolder service instance on the network in normal condition. If the QosManager fails to discover any PolicyHolder service, or if QosManager discovers more than one PolicyHolder service, the QosManager assumes the default policy of first-come-first serve admission control with default priorities based on 802.1D Annex G. The default value for the AdmissionPolicy is "Enabled".

The following table describes how the QosManager derives default priorities to be used in the Traffic Importance Number field. The QosManager takes the value of Traffic Class from the Tspec found inside the Traffic Descriptor structure, and looks up the corresponding default priority from the following table:

Traffic Class	**802.1D (Annex G) Traffic Type**	**Default Priority (Traffic Importance Number)**
Network Control	NC	7
Streaming Control	NC	7
Voice	VO	6
Gaming	VO	6
AV	VI	5
Audio	VI	5
Image	EE	3
Data	BE	0
Other	BE	0
Background	BK	1

In the context of UPnP AV, the AV Control Point invokes QosManager (from the UPnP AV Control Point) to perform the functions related to setting up Quality of Service a particular traffic. In case of non UPnP AV scenario, any application (acting as a Control Point) can invoke the QosManager service for setting up the Quality of Service for a particular traffic. In either case, the Control Point uses the 'QM:RequestTrafficQos' action exposed by the QosManager service to invoke the QosManager service. The control point puts the initial TrafficDescriptor as an input argument in the RequestTrafficQos action. For example, activeTspecIndex and parts of TrafficID (e.g. port numbers) may be unknown to the Control Point at the time of setup. These details may get filled in after the traffic QoS is implemented in the UPnP QoS Device. The control point may also

[2] QD: is QosDevice service.

[3] QM: is QosManager service

include: UserName, CpName, VendorApplicationName, PortName, and ServiceProviderServiceName. These fields are conveyed by the QoS Management entity to the QosPolicyHolder using the GetTrafficPolicy action. The QosPolicyHolder may use any information in the TrafficDescriptor to determine the TrafficImportanceNumber and UserImportanceNumber for the traffic stream that QoS is being requested for. It is recommended that the control point provides as much of information as available in the TrafficDescriptor XML fragment.

Upon receiving the 'QM:RequestTrafficQos' action, the QosManager Qos Management Entity obtains the TrafficImportanceNumber and the UserImportanceNumber for that TrafficDescriptor by querying the QosPolicyHolder. Traffic Importance Number (0-7) will be provided to QoS Device by the QosManager Qos Management Entity. User Importance Number (0-255) will be used by QosManager for traffic admission. In addition to User Importance Number and Traffic Importance Number, the QosPolicyHolder also returns a flag that indicates whether admission policy is enabled or disabled.

By invoking 'QD:GetPathInformation' action on QosDevice service instances on the network, the QosManager may determine which intermediate devices are on the path from the source to the sink. Some UPnP QosDevice service instances may also expose the 'QD:GetQosDeviceInfo' action. QoS management entity may invoke this action on the source and/or sink device to find out the port number and protocol information associated with that particular Traffic Descriptor. The QosManager Qos Management Entity issues the 'QD:GetQosState' action to every QoS Device on the path of the traffic stream and gets the dynamic status about that device. If QosManagement entity decides that there are insufficient resources, its may try another Tspec listed in the TrafficDescriptor. If QosManagement entity decides that there are enough resources available on all the devices on the path of the traffic, the QosManager issues 'QD:SetupTrafficQos' actions to those devices on the path. If SetupTrafficQos action is successfully executed on all the QoS Devices involved, then the updated TrafficDescriptor created by the QosManager is returned to the control point as an output argument to the 'QM:RequestTrafficQos' action. Otherwise, QosManager returns appropriate error in response to the 'QM:RequestTrafficQos' action.

QosStateId is a state variable defined in the QosDevice that enumerates the state of the device when it was queried by a QosManager with QD:GetQosState action. QosDevice returns QosStateId in response to QD:GetQosState as part of QosDeviceState argument. Subsequently, the QosManager repeats this string as input argument to QD:SetupTrafficQos. Please see QosDevice Service Definition for more details.

When a control point needs to either change or release the Quality of Service associated with a particular traffic, it uses the 'QM:UpdateTrafficQos' or 'QM:ReleaseTrafficQos' actions respectively. Upon receiving 'QM:UpdateTrafficQos' action, the QosManager repeats the admission control process described above for the revised TrafficDescriptor. Upon receiving 'QM:ReleaseTrafficQos' the QosManager invokes the 'ReleaseTrafficQos' action on all the devices on the path of that traffic.

A_ARG_TYPE_PathInformation state variable of QoS Device service is optional, but it is evented with moderation. Any time there is a change in Path information, the QoS device may issue an event and send the updated A_ARG_TYPE_PathInformation variable in the body of the event. QosManager may subscribe to these events. Upon receiving an event, QosManager may take an action that it deems necessary e.g. recomputing the topology.

Finally, it should be noted that UPnP QoS defines services (QosManager, QosDevice, and QosPolicyHolder) but it does not define new device type. Since Quality of Service problem needs to be solved across the board for multiple usage scenarios, it is expected that vendors may use any UPnP device as a container for the services defined in UPnP QoS. The control points and QoS Management Entities must look for UPnP QoS services embedded in all UPnP device types.

3. XML Service Description

```
<?xml version="1.0" encoding="utf-8"?>
<scpd xmlns="urn:schemas-upnp-org:QosManager:1">

   <specVersion>
      <major>1</major>
      <minor>0</minor>
   </specVersion>

   <actionList>
      <action>
         <name>RequestTrafficQos</name>
         <argumentList>
            <argument>
               <name>InitialTrafficDescriptor</name>
               <direction>in</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor
               </relatedStateVariable>
            </argument>
            <argument>
               <name>TrafficHandle</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficHandle
               </relatedStateVariable>
            </argument>
            <argument>
               <name>NumPolicyHolders</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_NumPolicyHolders
               </relatedStateVariable>
            </argument>
            <argument>
               <name>UpdatedTrafficDescriptor</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>BrowseAllTrafficDescriptors</name>
         <argumentList>
            <argument>
               <name>NumberOfTrafficDescriptors</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_NumTrafficDescriptors
               </relatedStateVariable>
            </argument>
            <argument>
               <name>TrafficDescriptorList</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_ListOfTrafficDescriptors
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>ReleaseTrafficQos</name>
         <argumentList>
            <argument>
               <name>RevokeTrafficHandle</name>
               <direction>in</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficHandle
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
      <action>
         <name>UpdateTrafficQos</name>
         <argumentList>
            <argument>
               <name>TrafficHandle</name>
               <direction>in</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficHandle
               </relatedStateVariable>
            </argument>
            <argument>
               <name>RequestedTrafficDescriptor</name>
               <direction>in</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor
               </relatedStateVariable>
            </argument>
            <argument>
               <name>ImplementedTrafficDescriptor</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_TrafficDescriptor
               </relatedStateVariable>
            </argument>
            <argument>
               <name>NumPolicyHolders</name>
               <direction>out</direction>
               <relatedStateVariable>A_ARG_TYPE_NumPolicyHolders
               </relatedStateVariable>
            </argument>
         </argumentList>
      </action>
   </actionList>

   <serviceStateTable>
      <stateVariable sendEvents="No">
         <name>A_ARG_TYPE_NumPolicyHolders</name>
         <dataType>ui4</dataType>
      </stateVariable>
      <stateVariable sendEvents="No">
         <name>A_ARG_TYPE_NumTrafficDescriptors</name>
         <dataType>ui4</dataType>
      </stateVariable>
      <stateVariable sendEvents="No">
         <name>A_ARG_TYPE_TrafficHandle</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="No">
         <name>A_ARG_TYPE_TrafficDescriptor</name>
         <dataType>string</dataType>
      </stateVariable>
      <stateVariable sendEvents="No">
         <name>A_ARG_TYPE_ListOfTrafficDescriptors</name>
         <dataType>string</dataType>
      </stateVariable>
   </serviceStateTable>
</scpd>
```

4. Test

No semantic tests have been specified for this service.

Annex A
(normative)

TrafficDescriptor XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
targetNamespace="http://www.upnp.org/schemas/TrafficDescriptorv1.xsd"
elementFormDefault="qualified" id="TrafficDescriptorv1">
 <xs:annotation>
   <xs:documentation xml:lang="en">
     TrafficDescriptor schema.
     Copyright 2004,2005 UPnP(tm). All rights reserved.
   </xs:documentation>
 </xs:annotation>
 <xs:element name="TrafficDescriptor" type="TrafficDescriptorType"/>
 <xs:complexType name="TrafficDescriptorType">
   <xs:sequence>
     <xs:element name="TrafficHandle" type="xs:string" minOccurs="0"/>
     <xs:element name="TrafficId">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="SourceAddress" type="IpAddress"/>
           <xs:element name="SourcePort" type="IpPortNumber"
minOccurs="0"/>
           <xs:element name="DestinationAddress" type="IpAddress"/>
           <xs:element name="DestinationPort" type="IpPortNumber"
minOccurs="0"/>
           <xs:element name="IpProtocol" type="IpProtocolType"
minOccurs="0"/>
           <xs:element name="v2TrafficId" type="v2TrafficIdType"
minOccurs="0"/>
           <!-- allow any element except those from target namespace -->
           <xs:choice minOccurs="0" maxOccurs="unbounded">
             <xs:any namespace="##other" processContents="lax"/>
             <xs:any namespace="##local" processContents="lax"/>
           </xs:choice>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
     <xs:element name="AvailableOrderedTspecList">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="Tspec" type="TrafficSpecification"
maxOccurs="unbounded"/>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
     <xs:element name="ActiveTspecIndex" type="xs:integer" minOccurs="0"/>
     <xs:element name="TrafficImportanceNumber" type="TrafficImportance"
default="0" minOccurs="0"/>
     <xs:element name="QosBoundarySourceAddress" type="IpAddress"
minOccurs="0"/>
     <xs:element name="QosBoundaryDestinationAddress" type="IpAddress"
minOccurs="0"/>
     <xs:element name="MediaServerConnectionId" type="xs:integer"
minOccurs="0"/>
     <xs:element name="MediaRendererConnectionId" type="xs:integer"
minOccurs="0"/>
     <xs:element name="TrafficLeaseTime" type="xs:integer" minOccurs="0"/>
     <xs:element name="v2" type="v2ExtensionType" minOccurs="0"/>
     <!-- allow any element except those from target namespace -->
     <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:any namespace="##other" processContents="lax"/>
       <xs:any namespace="##local" processContents="lax"/>
     </xs:choice>
     <xs:element name="OptionalPolicyParams" minOccurs="0">
       <xs:complexType>
         <xs:sequence>
           <xs:element name="UserName" type="xs:string" minOccurs="0"/>
           <xs:element name="CpName" type="xs:string" minOccurs="0"/>
           <xs:element name="VendorApplicationName" type="xs:string"
minOccurs="0"/>
           <xs:element name="PortName" type="xs:string" minOccurs="0"/>
           <xs:element name="ServiceProviderServiceName" type="xs:string"
minOccurs="0"/>
           <xs:element name="v2OptionalParams" type="v2OptionalParamsType"
minOccurs="0"/>
           <!-- allow any element except those from target namespace -->
           <xs:choice minOccurs="0" maxOccurs="unbounded">
             <xs:any namespace="##other" processContents="lax"/>
             <xs:any namespace="##local" processContents="lax"/>
           </xs:choice>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
   </xs:sequence>
 </xs:complexType>
 <xs:complexType name="TrafficSpecification">
   <xs:sequence>
     <xs:element name="TspecIndex" type="xs:integer"/>
     <xs:element name="AvTransportUri" type="xs:string" minOccurs="0"/>
     <xs:element name="AvTransportInstanceId" type="xs:integer"
minOccurs="0"/>
     <xs:element name="TrafficClass" type="TrafficClass" minOccurs="0"/>
     <xs:element name="v2TrafficSpecification"
type="v2TrafficSpecificationType" minOccurs="0"/>
     <!-- allow any element except those from target namespace -->
     <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:any namespace="##other" processContents="lax"/>
       <xs:any namespace="##local" processContents="lax"/>
     </xs:choice>
   </xs:sequence>
 </xs:complexType>
 <xs:simpleType name="TrafficClass">
   <xs:restriction base="xs:string">
     <xs:enumeration value="NetworkControl"/>
     <xs:enumeration value="StreamingControl"/>
     <xs:enumeration value="Voice"/>
     <xs:enumeration value="AV"/>
     <xs:enumeration value="Data"/>
     <xs:enumeration value="Audio"/>
     <xs:enumeration value="Image"/>
     <xs:enumeration value="Gaming"/>
     <xs:enumeration value="Other"/>
     <xs:enumeration value="Background"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="TrafficImportance">
   <xs:restriction base="xs:nonNegativeInteger">
     <xs:minInclusive value="0"/>
     <xs:maxInclusive value="7"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="IpPortNumber">
   <xs:restriction base="xs:nonNegativeInteger">
     <xs:minInclusive value="0"/>
     <xs:maxInclusive value="65535"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="IpProtocolType">
   <xs:restriction base="xs:nonNegativeInteger">
     <xs:minInclusive value="0"/>
     <xs:maxInclusive value="255"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="IPv6Address">
   <xs:restriction base="xs:hexBinary">
     <xs:length value="32"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="IPv4Address">
   <xs:restriction base="xs:string">
     <xs:pattern value="(([1-9]?[0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-
5])\.){3}([1-9]?[0-9]|1[0-9][0-9]|2[0-4][0-9]|25[0-5])"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:simpleType name="PrefixLength">
   <xs:restriction base="xs:positiveInteger">
     <xs:minInclusive value="1"/>
     <xs:maxInclusive value="128"/>
   </xs:restriction>
 </xs:simpleType>
 <xs:complexType name="IpAddress">
   <xs:sequence>
     <xs:choice>
       <xs:element name="Ipv4" type="IPv4Address"/>
       <xs:element name="Ipv6" type="IPv6Address"/>
     </xs:choice>
     <xs:element name="PrefixLength" type="PrefixLength" minOccurs="0"/>
   </xs:sequence>
 </xs:complexType>
 <xs:complexType name="v2ExtensionType">
   <xs:sequence>
     <xs:element name="PolicyHolderId" type="xs:string" minOccurs="0"/>
     <xs:element name="PolicyLastModified" type="xs:string" minOccurs="0"/>
     <xs:element name="PolicyModifyingUserName" type="xs:string"
minOccurs="0"/>
     <xs:element name="PolicyHolderConfigUrl" type="xs:string"
minOccurs="0"/>
     <xs:element name="v3" type="v3ExtensionType" minOccurs="0"/>
     <!-- allow any element except those from (v2) target namespace -->
     <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:any namespace="##other" processContents="lax"/>
       <xs:any namespace="##local" processContents="lax"/>
     </xs:choice>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v3ExtensionType">
   <xs:sequence>
     <xs:any namespace="##targetNamespace" processContents="lax"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v2TrafficIdType">
   <xs:sequence>
     <xs:element name="v3TrafficId" type="v3TrafficIdType" minOccurs="0"/>
     <!-- allow any element except those from (v2) target namespace -->
     <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:any namespace="##other" processContents="lax"/>
       <xs:any namespace="##local" processContents="lax"/>
     </xs:choice>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v3TrafficIdType">
   <xs:sequence>
     <xs:any namespace="##targetNamespace" processContents="lax"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v2OptionalParamsType">
   <xs:sequence>
     <xs:element name="v3OptionalParams" type="v3OptionalParamsType"
minOccurs="0"/>
     <!-- allow any element except those from (v2) target namespace -->
     <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:any namespace="##other" processContents="lax"/>
       <xs:any namespace="##local" processContents="lax"/>
     </xs:choice>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v3OptionalParamsType">
   <xs:sequence>
     <xs:any namespace="##targetNamespace" processContents="lax"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v2TrafficSpecificationType">
   <xs:sequence>
     <xs:element name="v3TrafficSpecification"
type="v3TrafficSpecificationType" minOccurs="0"/>
     <!-- allow any element except those from (v2) target namespace -->
     <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:any namespace="##other" processContents="lax"/>
       <xs:any namespace="##local" processContents="lax"/>
     </xs:choice>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
 <xs:complexType name="v3TrafficSpecificationType">
   <xs:sequence>
     <xs:any namespace="##targetNamespace" processContents="lax"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:anyAttribute/>
 </xs:complexType>
</xs:schema>
```

Description:

- TrafficHandle: String. Required. Every Traffic Descriptor on the network must have a unique traffic handle. When the value of trafficHandle is not known, it should be represented by a null string.
- TrafficID: The unique identifier for the traffic. Required. The traffic identifier may be defined via any combination of its components. No one single component is required. At least SourceAddress and DestinationAddress IP addresses must be specified in the TrafficId structure.
- SourceIP, DestinationIP: String (IP Address in the dotted decimal format). Optional.
- SourcePort, DestinationPort: TCP/UDP port number. Optional.
- IpProtocol: These are IANA assigned IP protocol numbers from 0-255. IGD trafficID defines IPProtocol in terms of these numbers. For more information: http://www.iana.org/assignments/protocol-numbers
- AvailableOrderedTspecList: Contains one or more Tspec components. The Tspec index reflects the order of preference. Tspecs with smaller index values are more preferred. availableOrderedTspecList is Required.

- Tspec: Tspec contains a description of Content QoS Requirements. In UPnP AV scenario, this information is extracted from the Content Directory Service of the Media Server. Tspec is represented in the form of an XML structure in the Content Directory Service. UPnP AV Control Point uses CDS:Browse and CDS:Search action calls to acquire the Tspec(s) associated with the content and inserts the Tspec XML into the Traffic Descriptor.

  Tspec XML structure consists of the following:

  - TspecIndex: Unique numerical index associated with a particular Tspec. The value of Tspec index indicates preference (as defined by the application or the end user). A Tspec with smaller Index is more preferred compared to a Tspec with larger Index.
  - AvTransportUri: Unique URI associated with the UPnP AV content item for which QoS is being requested. Mandatory only in case of three box UPnP AV model.
  - AvTransportInstanceId: Unique Instance ID associated with the UPnP AV Transport service associated with the content item for which QoS is being requested. Mandatory only in case of three box UPnP AV model.
  - TrafficClass: The traffic class associated with the traffic stream. This is an enumerated variable that can be assigned to one of the following list of values:
    - Network Control
    - Streaming Control
    - Voice
    - AV
    - Data
    - Audio
    - Image
    - Gaming
    - Other
    - Background

  TspecIndex must be filled in by the Control Point. Other Tspec parameters may be omitted if they are not known. Different Tspec(s) for the same traffic could be differentiated using traffic class. The following is a valid example of a Tspec list:

```
<Tspec>
   <TspecIndex>1</TspecIndex>
   <AvTransportUri>http://192.168.0.111/blah.mpg</AvTransportUri>
   <TrafficClass>AV</TrafficClass>
</Tspec>
<Tspec>
   <TspecIndex>2</TspecIndex>
   <AvTransportUri>http://192.168.0.111/blah2.mp3</AvTransportUri>
   <TrafficClass>Audio</TrafficClass>
</Tspec>
```

- ActiveTspecIndex: Integer index of active Tspec. Required when invoking QosDevice service. Optional when invoking QosManager.
- TrafficImportanceNumber: Traffic Importance (0-7) will be provided to QoS Device by the QosManager. In the absence of Policy Holder the QosManager will use default traffic class, as defined in Theory of Operation.
- QosBoundarySourceAddress: If a traffic stream originates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP QoS. Optional, because it is applicable only to traffic streams originating outside the home network. It should be noted that this address is not part of traffic identifier, because the IP packets will carry the IP address of the actual source address. QoS boundary point is used by QosManager for decisions related to path determination and device selection. The "Don't care" situation that may apply for SourceAddress IP address do not apply here. Hence a Control Point must not fill 0.0.0.0 for the value. If the SourceAddress is not in the subnet and a QosBoundarySourceAddress was not provided, the QosManager must return an error code 752.
- QosBoundaryDestinationAddress: If a traffic stream terminates outside the home network (e.g. on the Internet), the CP application must provide the QoS boundary point IP address. This address will be treated as the QoS termination point for UPnP Qos. Optional, because it is applicable only to traffic streams terminating outside the home network. It should be noted that this address is not part of traffic

identifier, because the IP packets will carry the IP address of the actual destination address. QoS boundary point is used by QosManager for decisions related to path determination and device selection. The "Don't care" situation that may apply for DestinationAddress IP address do not apply here. Hence a Control Point must not fill 0.0.0.0 for the value. If the DestinationAddress is not in the subnet and a QosBoundaryDestinationAddress was not provided, the QosManager must return an error code 752.

- OptionalPolicyParams: Various optional parameters can be provided. Optional.
- UserName: User Name field contains a string associated with the user who requested a particular traffic stream. This field, if present, shall identify the name of the user initiating the UPnP QoS Action. The field must be <= 64 UTF-8 characters. At present this specification does not define the semantics for this field. This field represent the name of the user initiating the QoS action. Examples include, "Dad", "Jimmy", "CN#1234567". The later is an exmple of a name provide by a service provider to uniquely identify its individual customers.

- CpName: CP Name field contains a string associated with the Control Point requesting the traffic stream. This field, if present, shall identify the name of the control point initiating the UPnP QoS Action. The field must be <= 64 UTF-8 characters. At present this specification does not define the semantics for this field. The CpName may specify the brand name of the control point or it may indicate the location of the control point such as "living room"

- VendorApplicationName: Vendor ApplicationName field contains a single URI string associated with an application. This field, if present, shall identify the name of the application initiating the UPnP QoS Action. For applications specified by vendors, the value of this field must begin with "urn:",followed by an ICANN domain name owned by the vendor, followed by ":application:", followed by an application name. i.e. the value of this field should be constructed as follows:

  urn:*domain-name*:application:*appl-name*

  domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

  appl-name: Name of the application provided by a vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The *appl-name* field must be <= 64 UTF-8 characters. The *application-name* must not contain the colon character.

  When requesting QoS it may be useful to know the manufacturer and name of the application software client or server that will process the traffic stream.

- PortName: Port Name field contains a single URI string associated with the fixed port used by an application for its connection set-up. This field, if present, shall identify the port number used by the application. The value of this field must begin with "urn:", followed by an ICANN domain name owned by the vendor, followed by ":port:", followed by the port name. i.e. the value of this field should be constructed as follows

  urn:*domain-name*:port:*port-number*

  domain-name: Application vendor domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

  port-number: A fixed port number used by the application either for connection set-up. This must be an integer value in the range 1 to 65535. If the application uses a range of port numbers then the starting port number of that range shall be specified. This value is either a vendor specific port used for the application or an IANA assigned port number for the application. The list of all the IANA assigned port numbers is maintained by IANA and is available at http://www.iana.org/assignments/port-numbers.

  These fileds indicate the port that is used by an application for its connection set-up. The use of IANA assigned port numbers is encouraged, but if not available for a particular application, a vendor assigned port number may be used.

- ServiceProviderServiceName: Service Provider Service Name field contains a single URI string associated with a service offered by a service provider. This field, if present, shall identify the name of the service offered by a service provider.. For services provided by service providers, the value of this field must begin with "urn:", followed by an ICANN domain name owned by a service provider , followed by “service:”, followed by a servicename. i.e. the value of this field should be constructed as follows:

urn:*domain-name*:service:*serv-name*

domain-name: Service Provider domain name. This must follow the syntax specified for Namespace Identifier (NID) in the RFC 2141. The *domain-name* field must be <= 64 UTF-8 characters.

serv-name: Name of the service provided by a service vendor. This must follow the syntax specified for Name Specific String (NSS) in the RFC 2141. The serv-*name* field must be <= 64 UTF-8 characters. The serv-*name* must not contain the colon character.

This represents the name of a particular service being offered by a service provider. Note that while multiple services provided by the same service provider may use the same software application, different Service offerings may be given different priorities (traffic importance).

- TrafficLeaseTime: Integer representing the lease time in seconds. Optional.
- MediaServerConnectionId: This is an optional field. The connection Id is obtained from the Media Server via CM::PrepareForConnection action. This field may be useful in case of UPnP AV based traffic streaming with multiple Tspecs. MediaServerConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details.
- MediaRendererConnectionId: This is an optional field. The connection Id is obtained from the Media Renderer via CM::PrepareForConnection action.This field may be useful in case of UPnP AV based streaming with multiple Tspecs. MediaRendererConnectionId may be used by the QosManager to identify the traffic stream that is being setup by the UPnP AV Control Point which is requesting Qos. Please refer to UPnP QoS Architecture document for more details.

Annex B
(normative)

Traffic Descriptor Matrix

Parameter Definition		Populated By			Used By[4]			Comments
Parameter Name	**R/ O**	**C P**	**Q M**	**Q D**	**C P**	**Q M**	**Q D**	
TrafficHandle	R		*		*	*	*	
TrafficID	R							
. SourceAddress	O	*			*	*		If SourceAddress tag is missing, it represents value not known. A value of 0.0.0.0 represents 'Don't Care' i.e. condition where CP wants to set up QoS for all traffic streams irrespective of the SourceAddress.
. SourcePort	O	*	*[5]				*	If SourcePort tag is missing, it represents value not known. This may happen in a '3 box model'.[6] A value of 0 represents 'Don't Care' i.e. condition where CP wants to set up QoS for all traffic streams irrespective of the SourcePort.
. DestinationAddress	O	*				*		If DestinationAddress tag is missing, it represents value not known. A value of 0.0.0.0 represents 'Don't Care' i.e. condition where CP wants to set up QoS for all traffic streams irrespective of the DestinationAddress.
. DestinationPort	O	*	*[7]				*	If DestinationPort tag is missing, it represents value not known. This may happen in a '3 box model'.[8] A value of 0 represents 'Don't Care' i.e. condition where CP wants to set up QoS for all traffic streams irrespective of the DestinationPort.

[4] Policy Holder may use any or all parts of the TD to make policy decisions. SincePolicy Holder is proprietary and policy definition is out of band for UPnP Qos; this table does not define what TD parameters are 'used by'Policy Holder.

[5] QosManager may query QD:GetQosDeviceInfo action call to obtain the port number.

[6] Please refer to the UPnP QoS Architecture document for more information around '3 box model'.

[7] QosManager may query QD:GetQosDeviceInfo action call to obtain the protocol number.

[8] Please refer to the UPnP QoS Architecture document for more information around '3 box model'.

. IpProtocol	O		*[9]				*	If Protocol tag is missing, it represents value not known. A value of 0 represents 'Don't Care' i.e. condition where CP wants to set up QoS for all traffic streams irrespective of the Protocol.
AvailableOrderedTspecList	R							
. Tspec	O							
. . TspecIndex	R	*				*	*	
..AvTransportUri	O	*			*	*		
..AvTransportInstanceId	O	*			*	*		
. . TrafficClass	R	*				*		
ActiveTspecIndex	R		*		*	*	*	
TrafficImportanceNumber	R		*				*	This information may come fromPolicy Holder, but is populated in TD by QosManager.
QosBoundarySourceAddress	O	*				*	*	For QosDevice to know if it is on path or not
QosBoundaryDestinationAddress	O	*				*	*	For QosDevice to know if it is on path or not
MediaServerConnectionId	O	*				*	*	Read architecture document section 5.1.2 for more information on how to use this value in setting up QoS in the case of a three-box AV model
MediaRendererConnectionId	O	*				*	*	Read architecture document section 5.1.2 for more information
UserName	O	*			*			
VendorApplicationName	O	*			*			
PortName	O	*			*			
ServiceProviderServiceName	O	*			*			
CpName	O	*			*			
TrafficLeaseTime	O	*					*	

[9] QosManager may query QD:GetQosDeviceInfo action call to obtain the IP Protocol.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

Tel: + 41 22 919 02 11
Fax: + 41 22 919 03 00
info@iec.ch
www.iec.ch